IN THE CLAIMS

1 (Currently Amended).   A method comprising:

assigning a number of stall cycles between a first and a second instruction; ~~and~~

determining a range within which one of said instructions may be reordered without violating data dependency;

reordering said instruction within said range; and

scheduling said first and second instructions for execution based on the assigned stall cycles.

2 (Previously Presented).   The method of claim 1, further comprising:

using a number of maximum possible pipeline stall cycles between said first and second instructions to indicate a data dependency therebetween.

3 (Original).   The method of claim 2, further comprising:

extending a register scoreboard that keeps track of the data dependency.

4 (Original).   The method of claim 3, further comprising:

maintaining a count of issue latency for said first and second instructions.

5 (Currently Amended).   The method of claim 3, further comprising:

maintaining a count for a number of cycles from start to end of [[a]] an issue of said first and second instructions.

6 (Original).   The method of claim 3, further comprising:

maintaining a count for pipeline stalls between said first instruction and a previous instruction.

7 (Currently Amended).   The method of claim 3, further comprising:

extending the register scoreboard ~~by m rows and m columns~~ to keep track of a maximum possible pipeline stall cycles.

8 (Previously Presented).   The method of claim 7, further comprising:

keeping track of a first non-zero value from right to left in a ~~an m-th~~ row of the register scoreboard to reorder said first instruction.

9 (Previously Presented).   The method of claim 7, further comprising:

keeping track of a first non-zero value from top to bottom in a ~~an m-th~~ column of the register scoreboard to reorder said first instruction.

10 (Original). The method of claim 3, further comprising:

keeping track of an instruction that causes pipeline stall.

11 (Currently Amended).   An apparatus comprising:

a register to store a number of stall cycles between a first and a second instruction; and

a compiler coupled to <u>determine a range within which one of said instructions may be reordered without violating data dependency, reorder said instruction within said range, and</u> schedule said first and second instructions for execution based on the stall cycles.

12 (Previously Presented).   The apparatus of claim 11, wherein said compiler uses a number of maximum possible pipeline stall cycles between said first and second instructions to indicate data dependency therebetween.

13 (Original). The apparatus of claim 12, wherein said register is extended by m-rows and m-columns to keep track of maximum possible pipeline stall cycles.

14 (Previously Presented).   The apparatus of claim 13, wherein said compiler to keep track of a first non-zero value from right to left in m-th row to reorder said first instruction.

15 (Previously Presented).   The apparatus of claim 13, wherein said compiler to keep track of a first non-zero value from top to bottom in the m-th column to reorder the first instruction.

4

16 (Currently Amended).   A system comprising:

        a non-volatile storage storing instructions;

        a processor to execute at least some of the instructions to provide a virtual machine that assigns a number of stall cycles between a first and a second instruction, determines a range within which one of said instructions may be reordered without violating data dependency, reorders said instruction within said range, and schedules said first and second instructions for execution based on the assigned stall cycles.

17 (Original). The system of claim 16, further comprising:

        a register to store dependency data between said first and second instructions.

18 (Previously Presented).   The system of claim 17, further comprising:

        a compiler coupled to schedule said first and second instructions for execution based on a maximum possible pipeline stall cycles.

19 (Currently Amended).   The system of claim [[16]] 17, wherein said register is a register scoreboard.

20 (Original). The system of claim 17, wherein said compiler is just-in-time compiler for an object-oriented programming language.

21 (Currently Amended).   An article comprising a computer readable storage medium storing instructions that, when executed cause a processor-based system to:

        assign a number of stall cycles between a first and a second instruction; and

        determine a range within which one of said instructions may be reordered without violating data dependency;

        reorder said instruction within said range; and

        schedule said first and second instructions for execution based on the assigned stall cycles.

22 (Currently Amended).   The article of claim 21, comprising a medium storing instructions that, when executed cause a processor-based system to:

use the number of maximum possible pipeline stall cycles between said first and second instructions to indicate a ~~the~~ data dependency therebetween.

23 (Original). The article of claim 22, comprising a medium storing instructions that, when executed cause a processor-based system to:

extend a register scoreboard that keeps track of the data dependency.

24 (Original). The article of claim 23, comprising a medium storing instructions that, when executed cause a processor-based system to:

maintain a count of issue latency for said first and second instructions.

25 (Currently Amended).   The article of claim 23, comprising a medium storing instructions that, when executed cause a processor-based system to:

maintain a count for a ~~the~~ number of cycles from start to end of the issue of said first and second instructions.

26 (Original). The article of claim 23, comprising a medium storing instructions that, when executed cause a processor-based system to:

maintain a count for pipeline stalls between said first instruction and a previous instruction.

27 (Original). The article of claim 23, comprising a medium storing instructions that, when executed cause a processor-based system to:

extend the register scoreboard by m rows and m columns to keep track of the maximum possible pipeline stall cycles.

28 (Currently Amended).   The article of claim 27, comprising a medium storing instructions that, when executed cause a processor-based system to:

keep track of a ~~the~~ first non-zero value from right to left in an ~~the~~ m-th row of the register scoreboard to reorder said first instruction.

6

29 (Currently Amended).   The article of claim 27, comprising a medium storing instructions that, when executed cause a processor-based system to:

keep track of a ~~the~~ first non-zero value from top to bottom in an ~~the~~ m-th column of the register scoreboard to reorder said first instruction.

30 (Original). The article of claim 23, comprising a medium storing instructions that, when executed cause a processor-based system to:

keep track of an instruction that causes pipeline stall.